# A Hybrid On-line Topic Groups Mining Platform

Cheng-Lin Yang[1] and Yun-Heh Chen-Burger[2]

[1] Centre for Intelligent Systems and their Applications, School of Informatics
University of Edinburgh, UK
s0969605@inf.ed.ac.uk
[2] School of Mathematical and Computer Sciences, Heriot-Watt University, UK
y.j.chenburger@hw.ac.uk

**Abstract.** In recent years, there is a rapid increased use of social networking platforms in the forms of short-text communication. Such communication can be indicative to popular public opinions and may be influential to real-life events. It is worth to identify topic groups from it automatically so it can help the analyst to understand the social network easily. However, due to the short-length of the texts used, the precise meaning and context of such texts are often ambiguous. In this paper, we proposed a hybrid framework, which adapts and extends the text clustering technique that uses Wikipedia as background knowledge. Based on this method, we are able to achieve higher level of precision in identifying the group of messages that has the similar topic.

**Keywords:** Social Network Analysis, Micro-blogging System, Machine Learning

## 1 Introduction

The information insides on-line communities can be very valuable for public options gathering or commercial marketing. Therefore, mining the user interested topic in large network such as Twitter becomes one of the most important task in social network analytics (SNA).

What is a topic group? Traditionally speaking, a topic group is a group of people who are gathered to embrace the same values or share the same responsibility. Moreover, with the rapid development of communication technology, the Internet has become an indispensable utility in daily life. People exchange information and knowledge on the Internet through various devices, forming a large social network and developing different types of on-line topic group. The user with new communication technology uses the forum or blog system to share his knowledge or experience with multimedia resources. He is able to discuss the topic with users from different countries by Internet. Therefore, a new type of community is formed. In this paper, we call them on-line topic groups.

Members of an on-line topic group are not restricted to the same geographical area unlike the topic group as defined in the traditional sense. An on-line topic group can be defined as a social phenomenon formed by a group of people who

communicate with each other through the Internet and share the same interest toward a certain topic. However, due to the short-length of the content used, the precise meaning and context of such texts are often ambiguous. To address these problems, we have devised a new topic mining approach that is an adaptation and extension of text clustering using Wikipedia as background knowledge.

## 2 Related Work

In this section, we discuss about community mining based on social network analysis approaches and based on other interesting approaches to see the overall perspective of community mining.

### 2.1 Social Network Analysis Approaches

According to [7], "Social network analysis studies social networks by means of analyzing structural relationships between people". Mining Community using traditional social network analysis approaches usually focuses on structure of the social network that is represented by direct or indirect graph. Each node in the graph represents an instance in the network e.g. person or object whereas links between nodes represent relations between the instances. The relations between the instances in the network can be defined by explicit information such as friends in Facebook or followers in Twitter.

[7] analyzed structure of network to identify communities among Slashdot users. Method of [7] is based on social network analysis approaches with negative weighted edges graph. A study of [1][3] also detected communities by considering the network structure. [3] took the network structure properties such as loops and edges of the network into the account. [1] considered bi-partite subgraphs to locate communities of websites.

### 2.2 Wikipedia Concepts Identification and Disambiguation

Recently, Wikipedia is used in many fields that are related to machine learning such as natural language processing, text classification and text clustering. One of difficulties for using Wikipedia is accurately matching between input text and Wikipedia concepts (articles) because each word or each phase in the input text can refer to one or more Wikipedia concepts. For example, a word "apple" can refer to both concepts "Apple (fruit)" and "Apple Inc.".

This problem has been interesting for a while and a lot of researchers have been proposed many methods to solve this problem. The review of word sense disambiguation methods can be found at [11].

An alternative approach that performs well in doing word sense disambiguation is using machine learning methods to learn labeled training set and classify ambiguous words. This concept is used in text annotation with Wikipedia links. [5][6][9] are several researches regarding to text annotation with Wikipedia links.

The first paper published about Wikipedia as a resource for annotation is [5] before significant improvement on this field by [6].

However, most papers did experiment in the context of standard length document. These experiments do not ensure that the approaches used in the papers will perform well in the case of short text document such as tweets, news or search snippets. In 2010, [9] brought the concept of annotating plain-text with Wikipedia links to context of short length document. They used anchor as a resource of identification instead of using only Wikipedia title as [8] because anchors are selected appropriately by people who create the pages. Their approach consists of three main steps: anchor parsing, anchor disambiguation and anchor pruning. Performing those steps extends an ability to deal with short text for the annotation system.

## 3 Proposed Hybrid Framework

A hybrid system with a three-layered framework: collection, classification and reasoning layers. The architecture of proposed system is shown in Figure 1.

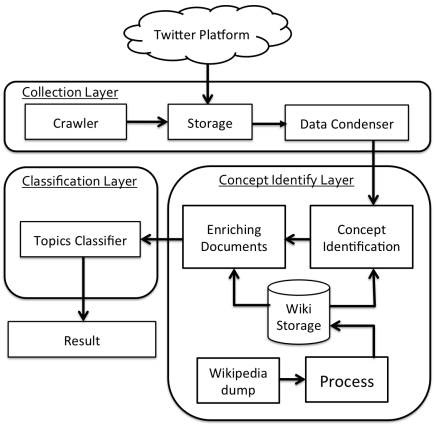


**Fig. 1.** Overview of proposed hybrid three-layered framework

### 3.1 Collection Layer

The collection layer contains components that fetch the data from the microblogging system, process the raw data into a pre-defined format and convert the data into numerical parameters.

**Crawler** The crawler is responsible for retrieving the user data from Twitter platform. All fetched tweets will be stored in the storage for further usage. It is designed to be a lightweight but targeted daemon so it can be deployed on multiple machines easily to increase the throughput.

**The Storage** In order to support fast lookup and flexible schema, the proposed system will take advantage of a distributed key-value database system: MongoDB [3], which allows us to change the table schema without altering the entire table. Scalability is another concern for any system that handles tremendous amount of data. A distributed database system provides a simple procedure to add new node into the system.

**Data Condenser** The data condenser reads the raw data from the database. The raw data contains noises like auxiliary words, emoticons or random characters, so, it is the data condenser's responsibility to remove these noises. It is also responsible for converting and normalising the selected fields such as steamed terms among tweets into numerical parameters for the concept identify layer.

### 3.2 Concept Indetify Layer

**Wikipedia Processing** We use 4th July 2012 English Wikipedia article dump which contains 4,012,083 articles and has a size about 8.2GB compressed. Then, we preprocessed and indexed them into 2 main catalogs to speed up the query.

1. Anchor Dictionary. We extracted all links and their anchors in Wikipedia pages and built them as anchor dictionary. The anchor dictionary is not like English dictionary. It contains only two important information: 1) anchors and 2) their corresponding Wikipedia concepts.
2. Wikipedia Pages. We also indexed Wikipedia pages content, their categories and inlink for speed in querying.

**Concept Identification** In order to identify Wikipedia concepts related to each tweet, we need to identify all anchors appearing in the tweet. In this subprocess, we use the steps given in Algorithm 1 to find the anchors. $lp(a)$ is link probability that can be calculated by following equation:

$$lp(a) = \frac{link(a)}{freq(a)} \tag{1}$$

where $link(a)$is number of anchor a used as a link and $freq(a)$ is number of anchor a appearing in all documents in collection.

[3] MongoDB: http://www.mongodb.org/

**Algorithm 1** Document Parsing

**Require:** input document $d$
  A =ngrams($d$, n=6)
  **for** each $word \in A$ **do**
    **if** $word \notin dictionary$ **then**
      A=A{$word$}
    **end if**
  **end for**
  **for** $a_1 \in A$ **do**
    **for** $a_2 \in A$ **do**
      **if** $a_1 \neq a_2$ and $substring(a_1, a_2)$ and $lp(a_1) < lp(a_2)$ **then**
        A=A{$a_1$}
      **end if**
    **end for**
  **end for**

**Concepts Disambiguation** Each anchor in the set of candidate anchors we got from previous sections could refer to several Wikipedia concepts. Therefore, in this step, we disambiguated those concepts and assigned the most appropriate concept to each anchor.

The same anchor may have different meanings and may link to different Wikipedia concepts depending on the context of the document. Therefore, we need the disambiguation process in order to select the appropriate Wikipedia concepts. We use a voting scheme adopted from [9]. The idea behind this voting scheme is that every anchor has to vote for all Wikipedia concepts related to other anchors in the document, except concepts related to itself. Then, the concepts that are given a top e-rank by their voting score are selected as candidate concepts. Finally, we select the most appropriate concept by using a commonness score. The detail of this voting scheme is described as follows:

First, we calculate relatedness between two Wikipedia concepts by using Normalized Google distance $rel(p_a, p_b)$ between the inlink of $p_a$ and $p_b$ where $p_b$ is the Wikipedia concept corresponding to anchor $b$.

Next, we calculate the voting score of anchor b to concept pa by averaging the relatedness between all corresponding concepts of anchor b to concept pa, with prior probability known as commonness $Pr(p_b|b)$ as shown in the following equation:

$$vote_b(p_a) = \frac{\sum_{p_b \in P_g(b)} rel(p_a, p_b) \cdot Pr(p_b|b)}{\|P_g(b)\|} \tag{2}$$

After that, the total score assigned to $p_a$ can be calculated by

$$rel_a(p_a) = \sum_{b \in A\{a\}} vote_b(p_a) \tag{3}$$

Using only this score to assign the concepts to the anchors may not be enough because, as mentioned in [6], balancing the score and commonness is the main factor affecting the performance.

**Concepts Filtering** However, after the disambiguation step is applied, there may be uncorrelated concepts left. Therefore, we have a final concepts filtering step to remove all concepts that are not related to others. We filter out unrelated anchors by using the concept of coherence between selected concepts from the concepts disambiguation step. To calculate the coherence score, we use the average relatedness between selected concepts as follows:

$$coherence(a \rightarrow p_a) = \frac{1}{\|S\| - 1} \sum_{p_b \in S\{p_a\}} rel(p_a, p_b) \tag{4}$$

where $S$ is number of all selected concepts. Next, we filter out unrelated anchors that satisfy the following condition:

$$\frac{coherence(a \rightarrow p_a) + lp(a)}{2} < \epsilon \tag{5}$$

where $\epsilon = 0.2$ and $lp(a)$ is the link probability of an anchor $a$.

**Document Enriching** Most terms in the tweets tend to appear only once. With these characteristics of short text document, there are some problems with TF-IDF weighting that we use for modeling the document. In some cases, important words will have lower inverse document frequency compared to those who are not important. This means, in some cases, important words will have lower inverse document frequency compared to those who are not important.

[2] succeed in using 3 different strategies to enrich TF-IDF vector with background knowledge based on WordNet. Three strategies consist of adding corresponding WordNet concepts, replacing terms by WordNet concepts and replacing term vector with concept vector. Also, [8] yielded the good results from enriching the document with semantic related terms based on Wikipedia knowledge. In our proposed method, we adapted the strategies from [2] and [8] to enrich Wikipedia knowledge into the tweets.

*Strategy 1: Add Wikipedia concepts* We replace and add terms in each tweet with its corresponding Wikipedia concepts. The reason is that one of the problems that reduce the performance of bag-of-word model is that each Wikipedia concept can be mentioned by several different words/phrases. This problem leads to low cosine similarity between these two documents because the word "MS" and "Microsoft" are treated as different words. Therefore, to reduce the error of different representation of the same concept, we replace them with Wikipedia concept which change them into the same representation.

*Strategy 2: Add Wikipedia concepts and categories* We extend the first strategy by adding categories of Wikipedia concepts corresponding to each term in a tweet because another problem of bag-of-word model is that the model cannot capture semantic relationship between two related terms. Adding Wikipedia categories will solve the problem of semantic relationship between the tweets.

Adding Wikipedia categories can solve the problem semantic relationship of bag-of-word model. Therefore, in this strategy, we also add Wikipedia concepts in documents besides replacing and adding Wikipedia concepts in Strategy 1.

### 3.3 Classification Layer

The results from many papers show that affinity propagation is the best among several clustering algorithms in short text clustering task. However, there are many criticism of affinity propagation about a problem with a large dataset. The issue about scalability of affinity propagation was mentioned in [10, 13]. Therefore, due to the size of our dataset, we decided to use Bisecting k-means clustering because of its scalability and efficiency.

## 4 Evaluation

The aim of this paper is mining groups in Twitter by finding a group of tweets which have the same concepts. Our approaches are based on text clustering and integration of Wikipedia knowledge and vector space model.

### 4.1 Experiments

In order to evaluate our methods, we did three experiments as follow and applied them to 1,500,000 collected tweets:

- *[Experiment 1]* In order to evaluate our work, we need some approaches to be compared with. Therefore, before experiment on our approaches, we ran experiment on pure clustering algorithm without enriching Wikipedia knowledge.
- *[Experiment 2]* In this method, we did further preprocessing step with our tweets data by adding related Wikipedia concepts as we explained in Strategy 1. After that, we model those enriched tweets with TF-IDF vectors before clustering.
- *[Experiment 3]* This method extended the concepts from Method 2. It does not only add Wikipedia concepts that related to each tweets in the preprocessing step but also add Wikipedia categories of each Wikipedia concepts into the tweets in order to solve the semantic relation problem of bag-of-word model

### 4.2 Evaluation based on ground-truth testset

For evaluation based on ground-truth testset, we manually labelled 400 tweets with appropriate groups. Then, we evaluate all three methods we mentioned earlier by calculating V-Measure score between true labels and labels that were assigned by clustering algorithm.

**Evaluation Metric** Typically, the basic criteria of a clustering result are homogeneity and completeness. The homogeneity criterion is satisfied, for all clusters, every member of each cluster comes from only one class which is defined as:

$$homogeneity = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise} \end{cases} \tag{6}$$

where $H(C|K)$ is the conditional entropy of the classes given assigned clusters and $H(C)$ is the entropy of the class defined as:

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{n_{c,K}}{n}\log\frac{n_{c,k}}{\sum_{c=1}^{|C|}n_{c,k}} \tag{7}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}n_{c,k}}{n}\log\frac{\sum_{k=1}^{|K|}n_{c,k}}{n} \tag{8}$$

in which $n$ is number of all data points and $n_{c,k}$ is number of data points from class $c$ that clustered into cluster $k$.

Completeness criterion is quite opposite to homogeneity. It is satisfied if all members of a class are clustered into the same cluster. Mathematically, we can define completeness score as follow:

$$completeness = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise} \end{cases} \tag{9}$$

where $H(K|C)$ is the conditional entropy of assigned clusters given the classes and $H(K)$ is the entropy of assigned clusters defined as:

$$H(K|C) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{n_{c,K}}{n}\log\frac{n_{c,k}}{\sum_{k=1}^{|K|}n_{c,k}} \tag{10}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}n_{c,k}}{n}\log\frac{\sum_{c=1}^{|C|}n_{c,k}}{n} \tag{11}$$

in which $n$ is number of all data points and $n_{c,k}$ is number of data points from class $c$ that clustered into cluster $k$.

A good clustering result should satisfy both homogeneity and completeness at the same time. In order to do that, we used V-Measure as a metric for evaluating clustering results. V-Measure which is first introduced by [12] is the harmonic mean of homogeneity and completeness scores bounded in the range of [0, 1] . The closer the value is to 1, the better the quality of a clustering result. It can be defined as the following equation:

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta + h) + c} \tag{12}$$

where $\beta$ is the weight, if $\beta$ is set less than 1, homogeneity is weighted more. If $\beta$ sets to more than 1, completeness is weighted more. In our experiment, we weight them equally by setting $\beta = 1$.

### 4.3 Results

For each experiment, we ran bisecting k-means multiple times with different setting up of number of clusters $k$ ranging from 64 to 4096. Then, we evaluate the clustering results with our testset using V-Measure as an evaluation metric. Figure 2 shows V-Measure scores of Experiment 1, 2 and 3 with different setting of number of clusters $k$.

From the figure, it is clear to see that *Method 2 (concepts)* and *Method 3 (concepts+categories)* outperformed *Method 1 (baseline)*. *Method 2* is clearly better than *Method 1* at every setting of number of clusters $k$. The highest V-Measure that *Method 1* can get is 0.674 at $k = 3800$ whereas the highest V-Measure of *Method 2* is 0.747 at $k = 3400$. The difference between the highest peak of them is 7.3%.

Furthermore, in the case of *Method 3*, it has clearly higher performance than *Method 1* in the Figure 2. Comparing with their best performance, *Method 3* gets 14.7% better with V-Measure 0.821 at $k = 3600$. We can conclude from these results getting from the testset that *Method 2* and *Method 3* have dramatic improvement from *Method 1* with V-Measure 0.747, 0.821 and 0.674. From these results, it confirms that using Wikipedia as a resource for enriching the tweets can improve the performance of topic groups mining.

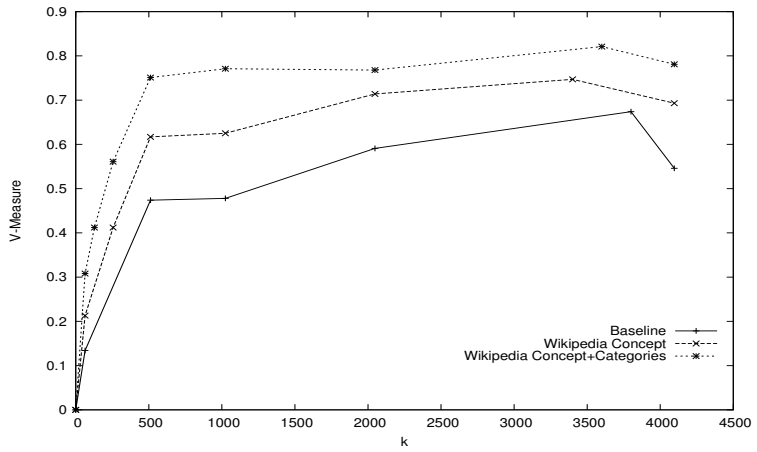


**Fig. 2.** Comparison among Method 1,2 and 3

## 5 Conclusion

In this paper, we proposed a state-of-the-art three layered framework to mine topic groups from large scale short text documents (Tweets) based on Wikipedia knowledge. Mining topic groups based on social network analysis approaches fails

to capture hidden concepts in a network because they usually model the network as a graph with explicit relationships that can be found in the network connectivity. The hidden concepts in the network are not taken into account leading these approaches to missing the important concepts among users in the network. In this study, we therefore investigated an alternative approach to identify hidden concepts in Twitter. We used clustering based methods to mine hidden concepts in tweets based on their topics. The optimistic result we have is the method enriched with Wikipedia concept and categories. Based on the evaluation metric in Section 4.3, our method has a promising V-measure score up to 0.821 from real Twitter data where baseline method only has 0.674 in V-measure score.

## References

1. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1481–1493, NY, USA, 1999.
2. A. Hotho, A. Hotho, S. Staab, S. Staab, G. Stumme, and G. Stumme. Text clustering based on background knowledge, 2003.
3. M. E. J. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330, Mar. 2004.
4. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *In CIKM 07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
5. D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, NY, USA, 2008. ACM.
6. J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 741–750, NY, USA, 2009. ACM.
7. P. Wang, J. Hu, H.-J. Zeng, and Z. Chen. Using wikipedia knowledge to improve text classification. *Knowl. Inf. Syst.*, 19(3):265–281, May 2009.
8. P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1625–1628, NY, USA, 2010. ACM.
9. Y. Fujiwara, G. Irie, and T. Kitahara. Fast algorithm for affinity propagation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2238–2243. AAAI Press, 2011.
10. R. Navigli. A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of the 38th international conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM'12, pages 115–129, Berlin, Heidelberg, 2012.
11. A. Roseberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure.
12. X. Zhang, C. Furtlehner, and M. Sebag. Distributed and incremental clustering based on weighted affinity propagation. In Proceedings of the 2008 conference on STAIRS 2008, pages 199–210, Amsterdam, The Netherlands, 2008. IOS Press.